

Amendments to the Claims:

1. (Currently Amended) A ~~communications~~ system comprising:  
a first host ~~capable of transmitting~~ configured to transmit multiplexed data at a first data transfer rate;  
a second host ~~capable of receiving~~ configured to receive multiplexed data at a second data transfer rate;  
a network through which data is transferred from the first host to the second host having a third data transfer rate; and  
a data throttle, wherein the data throttle limits the first data transfer rate to a throttle value that is less than or equal to the least one of the first data transfer rate, the second data transfer rate, and the third data transfer rate, and wherein the first, second, and third data transfer rates are obtained during a communication start-up process from a signaling message.
- 2-3. (Canceled).
4. (Original) The system of Claim 1, wherein the throttle value is a predetermined value.
5. (Original) The system of Claim 1 wherein the first host is further comprised of an applications layer, a sockets layer, a transport layer, and a network layer.
6. (Previously Presented) The system of Claim 5, wherein the data throttle operates by one or more application program interface (API) calls from the applications layer to the sockets layer, said API calls limiting the transmission data rate to the throttle value.

7. (Original) The system of Claim 5, wherein the transport layer is comprised of a User Datagram Protocol (UDP) and the network layer is comprised of an Internet Protocol (IP).

8-11. (Canceled)

12. (Previously Presented) The system of Claim 1, wherein the communications start-up process is a Session Initiation Protocol (SIP) process.

13-15. (Canceled)

16. (Previously Presented) A method comprising:  
obtaining a data transfer rate of a first host and a data transfer rate of a second host at which the second host may receive data;

obtaining a data transfer rate of respective one or more intervening packet processing platforms located between the first and second host, wherein the data transfer rate of the first host, data transfer rate of the second host, and data transfer rate of the respective one or more intervening packet processing platforms are obtained during a communication start-up process from a signaling message;

setting, during a communication start-up process, a throttle value that is less than or equal to the least of the data transfer rate of the first host, the data transfer rate of the second host, and the data transfer rate of respective one or more intervening packet processing platforms located between the first and second hosts; and

transmitting data packets from the first host to the second host at a data transfer rate that is less than or equal to the throttle value.

17. (Original) The method of Claim 16, wherein setting the maximum data transfer rate of the first host to the throttle value is accomplished by Application Programming Interface (API) calls from an application executing on the first host to a sockets layer of the first host.

18. (Original) The method of Claim 16, wherein transmitting data packets from the first host to the second host at a data transfer rate that is less than or equal to the throttle value is accomplished by use of a User Datagram Protocol (UDP) transport layer and an Internet Protocol network layer.

19. (Previously Presented) A method comprising:  
receiving, during a communication start-up process, a throttle value that is less than or equal to the least of a data transfer rate of a source device, a data transfer rate of a destination device, and a data transfer rate of respective one or more intervening packet processing platforms located between the source and destination devices, wherein the data transfer rate of the source device, data transfer rate of the destination device, and data transfer rate of the respective one or more intervening packet processing platforms are obtained during a communication start-up process from a signaling message;  
setting the maximum data transfer rate of the source device to the throttle value; and  
transmitting data packets from the source device to the destination device at a data transfer rate that is less than or equal to the throttle value.

20. (Previously Presented) The method of Claim 19, wherein setting the maximum data transfer rate of the source device to the throttle value is accomplished by Application Programming Interface (API) calls from an application executing on the source device to a sockets layer of the source device.

21. (Previously Presented) The method of Claim 19, wherein transmitting data packets from the source device to the destination device at a data transfer rate that is less than or equal to the throttle value is accomplished by use of a User Datagram Protocol (UDP) transport layer and an Internet Protocol network layer.

22. (Previously Presented) An apparatus comprising:  
a processor and a memory storing executable instructions that in response to execution by the processor cause the apparatus to at least perform the following:

obtain a data transfer rate of a first host, a data transfer rate of a second host, and a data transfer rate of a network through which data is transferred from the first host to the second host during a communication start-up process from a signaling message; and

set a throttle value for transmission of data from the first host to the second host to a value that is less than or equal to the least of the data transfer rate of the first host, the data transfer rate of the second host, and the data transfer rate of the network, wherein the throttle value is set during a communication start-up process.

23-25. (Canceled).

26. (Previously Presented) The apparatus of Claim 22, wherein the processor is further configured to:

set the maximum data transfer rate of the first host to the throttle value.

27. (Previously Presented) The apparatus of Claim 26, wherein in order to set the maximum data transfer rate of the first host to the throttle value, the processor is further configured to:

establish one or more Application Programming Interface (API) calls from an application layer of the first host to a socket layer of the first host.

28. (Canceled).